IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Ilya Kirnos

Group Art Unit No.: 2143

Examiner: SHIN, KYUNG H.

Serial No.: 09/866,143

Filed: May 25, 2001

For: MANAGEMENT AND SYNCHRONIZATION APPLICATION FOR NETWORK FILE SYSTEM

Mail Stop Appeal Briefs-Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed on April 4, 2007.

I. REAL PARTY IN INTEREST

Oracle Corporation is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

Appellant is unaware of any related appeals and interferences.

III. STATUS OF CLAIMS

Claims 1-33, 76-88, and 90-107 are pending in the application.

Claims 1-3, 6-9, 12-16, 21, 24, 25, 31-33, 76-78, 81, 84, 87-88, 90, 95, 99, and 105-10722-24 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Verma et al. (U.S. Patent No. 6,856,993) in view of Bailey et al. (U.S. Patent No. 6,473,767, and further in view of Rudoff (U.S. Patent No. 6,636,878).

Claims 4, 5, 7, 8, 10, 11, 17-20, 22, 23, 26-30, 79, 80, 82, 83, 85, 86, 91-94, 96-98, 100-104 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Verma-Bailey-Rudoff and further in view of Wolff (U.S. Patent No. 6,101,508).

It is from this final rejection of Claims 1-33, 76-88, and 90-107 that this Appeal is taken.

IV. STATUS OF AMENDMENTS

The Claims have not been amended after the Final Office Action.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The summary below includes references to material within the Specification. These references are not meant to be exhaustive.

Claim 1 recites a computer-implemented method for managing files, the method comprising: (See Specification, Page 5, lines 1-5).

recording information about one or more items in a file system to a comparison file, wherein the information recorded to the comparison file includes location information to identify where in the file system the one or more items are located; (See Specification, Page 4, lines 3-12).

generating a working version of a portion of the file system, the working version including at least one or more working items that correspond to the one or more items located in the file system; (See Specification, Page 4, lines 3-12).

persistently maintaining the working version; (See Specification, Page 3-4, lines 17-5).

upon a synchronization event, comparing the location information for the one or more items in the comparison file to the working version to determine if any of the corresponding one or more working items has been moved to a new location in the working version. (See Specification, Page 5, lines 15-25).

Claim 12 recites a computer-implemented method for managing files, the method comprising: (See Specification, Page 5, lines 1-5).

recording information about one or more items in a file system to a comparison file, wherein the information recorded to the comparison file includes location information to identify where in the file system the one or more items is located; (See Specification, Page 4, lines 3-12).

making a working version of a portion of the file system, wherein the working version includes one or more working items corresponding to the one or more items in the file system; and (See Specification, Page 4, lines 3-12).

persistently maintaining the working version; and (See Specification, Page 3-4, lines 17-5).

upon a synchronization event, comparing information in the comparison file to the working version to determine if a first working item in the working version was copied from a second working item in the working version. (See Specification, Pages 5-6, lines 25-2).

Claim 25 recites a computer-implemented method for managing files, the method comprising: (See Specification, Page 5, lines 1-5).

recording information about a file system to a comparison file, the information identifying at least a first item in the file system located at a first location, the first location being identifiable by a first location information; (See Specification, Page 4, lines 3-12).

making a working version of a portion of the file system, the working version including at least a first working item originating from the first item, the first working item initially being located at a first working location, the first working location corresponding to the first location information; (See Specification, Page 4, lines 3-12).

persistently maintaining the working version; and (See Specification, Page 3-4, lines 17-5).

upon a synchronization event, comparing information in the comparison file to the working version to detect if at least one of two operations were performed on the working version, the operations including changing the first working location information for the first working item, and editing a content of the first working item. (See Specification, Page 8, lines 15-25).

Claim 31 recites a computer-implemented method for managing files, the method comprising: (See Specification, Page 5, lines 1-5).

recording information about a file system to a comparison file, the information identifying at least a first item in the file system located at a first location, the first location being identifiable by a first location information; (See Specification, Page 4, lines 3-12).

making a working version of a portion of the file system, the working version including at least a first working item originating from the first item, the first working item initially being located at a first working location, the first working location corresponding to the first location information; (See Specification, Page 4, lines 3-12).

persistently maintaining the working version; and (See Specification, Page 3-4, lines 17-5).

upon a synchronization event, comparing information in the comparison file to the working version to determine if a compound operation was performed on the first working item, the compound operation including at least two successive operations from a set of operations that consist of changing the first working location information for the first working item, making a first working copy from the first working item, and editing a content of the first working item. (See Specification, Page 8, lines 15-25).

Claims 76, 87, 99, and 105 are computer readable storage medium reasonably analogous to Claims 1, 12, 25, and 31. (See Specification, Page 32-33, lines 25-14).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether Claims 1-3, 6-9, 12-16, 21, 24, 25, 31-33, 76-78, 81, 84, 87-88, 90, 95, 99, and 105-107 are patentable over Verma in view of Bailey and further in view of Rudoff.

Whether Claims 4, 5, 7, 8, 10, 11, 17-20, 22, 23, 26-30, 79, 80, 82, 83, 85, 86, 91-94, 96-98, 100-104 are patentable over Verma-Bailey-Rudoff and further in view of Wolff

ARGUMENTS

Claims 1-3, 6, 9, 12-16, 21, 24, 25, 31-33, 76-78, 81, 84, 87-90, 95, 99 and 105-107 were rejected under 35 U.S.C. § 103(a), as being unpatentable over U.S. Pat. No. 6,856,993 to Verma et al. ("Verma") in view of U.S. Patent No. 6,473,767 to Bailey ("Bailey") and further in view of U.S. Patent No. 6,636,878 to Rudoff ("Rudoff"). Claims 4, 5, 7, 8, 10, 11, 17-20, 22, 23, 26-30, 79, 80, 82, 83, 85, 86, 91-94, 96-98, and 100-104 were rejected under 35 U.S.C. §103(a) as being unpatentable over Verma in view of U.S. Pat. No. 6,101, 508 to Wolff ("Wolff").

I. EACH OF CLAIMS 1-3, 6-9, 12-16, 21, 24, 25, 31-33, 76-78, 81, 84, 87-88, 90, 95, 99, AND 105-107 IS PATENTABLE OVER ANY ALLEGED COMBINATION OF VERMA IN VIEW OF BAILEY AND FURTHER IN VIEW OF RUDOFF

As will be seen from the discussion below, the Examiner's rejections of 1-3, 6-9, 12-16, 21, 24, 25, 31-33, 76-78, 81, 84, 87-88, 90, 95, 99, and 105-107 under 35 U.S.C. § 103(a) are improper because the cited references Verma, Bailey, and Rudoff do not teach all of the limitations of those claims. Moreover, there are clear errors of fact and of law in the Examiner's rejection that make the rejection improper and without basis.

A. Claim 1

MPEP § 706.02(j) states that in order to establish a *prima facie* case of obviousness, one of the three basic criteria that must be met is that "the prior art reference (or references when combined) must teach or suggest all the claim limitations." In rejecting Claim 1, the **Examiner fails to show that any of the cited references,**

individually or in combination, teach or suggest at least one of the claim limitations of Claim 1.

Claim 1 recites:

> A computer-implemented method for managing files, the method comprising:
> **recording information about one or more items in a file system to a comparison file,** wherein the information recorded to the comparison file includes location information to identify where in the file system the one or more items are located;
> generating a working version of a portion of the file system, the working version including at least one or more working items that correspond to the one or more items located in the file system;
> persistently maintaining the working version; and
> **upon a synchronization event, comparing the location information for the one or more items in the comparison file to the working version to determine if any of the corresponding one or more working items has been moved to a new location in the working version.**

As recited in Claim 1, a method is provided to improve synchronization operations between an original version of a file system and a copy of the file system. Conventionally, synchronization operations often involved a significant amount of resource consumption. For example, conventionally, to synchronize files, an application would detect the addition of a new item to a copy of a file system and then transfer the entire contents of the new file back to the original file system. (See Specification, Page 2).

This is inefficient, especially if a new item in the copy of the file system is merely a copy of an existing item in the file system. For example, a user moves an item from one location in a copy of the file system to another would, upon synchronization; the original file system would delete the original item and create a new item in the new location. These operations are computationally expensive. (See Specification, Page 2). Claim 1 provides a mechanism that overcomes many of these performance issues.

To do so, Claim 1 recites "**recording information about one or more items in a file system to a comparison file**" and "**upon a synchronization event, comparing the location information for the one or more items in the comparison file to the working version to determine if any of the corresponding one or more working items has been moved to a new location in the working version.**"

Such a method is not taught or suggest by Verma, Bailey, or Rudoff, either individually or in any alleged combination thereof.

1. Verma

First of all, as discussed below, Verma stores the results of operations, not information about a file at the creation of a working version of a file system. Moreover, the Verma results are never stored in a manner that would allow them to be called upon in a subsequent comparison between a file system and a working version of the file system. Basically, Verma does not record information about a file to a comparison file.

Verma discusses techniques for creating a transactional file system, which allows application transactions to be committed to disk without compromising data integrity. (See Verma, Col. 2, lines 12-22). To do so, Verma defines a transaction coordinator that enlists and maintains a set of resource managers to control how and when specific commands are executed and how and when the results of those commands are committed to disk. (See Col. 8, lines 16-30).

The Examiner argues that "[a] transaction being performed on a file system is a change to that file system." (See Final Office Action, Page 2). The Examiner also states that "Verma discloses that information concerning the changes to a file system are stored", that "this stored information is equivalent to a comparison file", and that the completion of a transaction on a file system synchronizes that file system with the required file system update information." (See Final Office Action, Page 2). This is not true. Verma uses a transactional context object to store file system **operations**, such as file open, create file, close file, etc., that are to be performed by a particular transaction. (See Col. 2, lines 12-33). Information about the file system itself is not recorded to the transactional context object. Accordingly, changes to the file system are made by performing a series of operations stored in the transaction context object, and then committing the changes to disk. Throughout this process, Verma **does not** record information about files to a comparison file, and Verma **does not** use the comparison file to determine what changes have occurred in a file system. It is the operations in the transactional context object define those changes.

In the Final Office Action, the Examiner cites Verma Action (Col. 9, lines 10-17) as the section that teaches "recording information". That section of Verma specifically references a file handle returned in response to a successful create/open request that will include a pointer to the transaction context. The Appellants are unclear as to which of these items (e.g., file handle or transaction context) are the supposed equivalents of the comparison file. The Appellants assume it is the transaction context since it at least includes information regarding a particular transaction. As noted above, the transaction context is an object (See *Verma*, Col. 7, line 8; Col. 7, lines 31-32; Col. 7, line 53) that contains information regarding <u>a particular operation</u> being performed on a file system. Once a Verma operation is committed, the transaction context operation is executed. Each transaction context is executed in turn. For example, if one transaction context writes a file to the file system and a subsequent object deletes the file, Verma performs each of these operations in turn. A set of transaction contexts is not the equivalent of a comparison file.

Moreover, Verma does not "compare the location information for the one or more items in the comparison file to the working version to determine if any of the corresponding one or more working items has been moved to a new location in the working version." Verma makes absolutely no mention of comparing information in a comparison file to a working version of a file system in order to determine whether changes in the working version need to be made in the original file system to maintain system synchronization.

In other words, Verma cannot perform the reads and writes by performing a comparison operation. To do so, Verma would have to be modified in a number of ways. First, it would have to be modified to examine transaction contexts individually and as groups in order to derive all of the changes that might have occurred in the file system. This would add the unnecessary overhead the claimed invention avoids. In other words, the Verma system would have to examine each transaction context to determine what operation was performed and how it affected the file system. Then it would have to determine whether a subsequent transaction context overwrote a previous transaction context, whether a prior transaction context contains the original information about the file system, etc. And even if Verma did this type of analysis, this analysis is still not the same as "**recording information about one or more items in a file system to a comparison file**". Nor would that Verma analysis "**compare the location information for the one or more items in the comparison file to the working version to determine**

if any of the corresponding one or more working items has been moved to a new location in the working version."

Accordingly, the Appellants believe that the Examiner's argument that the way Verma stores file information is the equivalent of what is described in Claim 1 has been overcome.

2. Bailey

Bailey does not teach or suggest "recording location information to a comparison file" or "comparing the location information for the one or more items in the comparison file to the working version." Bailey describes a system that creates two directories, a source and a target directory, that are compared after file system operations have been performed on one of the two directories. (See Bailey, Col. 2, lines 41-56 and Col. 4, lines 6-18).

To make the comparison, Bailey inserts "anti-files" into its directories to indicate when a file has been removed from one of the two directories. When an operation requires the two directories to be synched up, the two directories are compared directly to each other, where an anti-file has been inserted in one of the directories, the Bailey system determines whether it should delete the corresponding file in the other directory. (See Bailey, Col. 2, lines 41-56). To synch the directories, the two directories are compared directly to each other. When an anti-file is encountered the corresponding file is removed. Bailey does not record information to a separate comparison file, and any comparison between the source and target directories in Bailey is performed directly between the two.

Claim 1 avoids comparing two file systems directly by recording information to a comparison file. Accordingly, Bailey does not teach or suggest at least those two elements of Claim 1.

3. Rudoff

Rudoff also lacks any teaching or suggestion that would indicate recording information to a comparison file and comparing information from the comparison file to a working version of a file system. Rudoff does not teach a "persistently maintaining the working version of a file system" as the Examiner indicates. (See Final Office Action, Page 6). Instead, Rudoff describes "a mechanism for replicating and maintaining files in a space efficient manner using a copy-on-write approach" (See Rudoff, Col. 4, lines 50-60). Rudoff does this in order to optimize space usage on a disk drive.

Importantly, the Examiner makes no allegation that Rudoff does teach the recording or comparing limitations of Claim 1. Therefore, even if a combination of Verma, Bailey, and Rudoff were obvious (although the Appellant disputes such a claim), together they fail to teach or suggest at least the first and last elements of Claim 1.

4. There is No Motivation to Combine

Furthermore, there is no motivation to combine Verma, Bailey, and Rudoff. The Examiner first argues that it "would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Verma to enable the usage of a comparison file with file directory entries for processing as taught by Bailey." The Appellants respectfully disagree. First, Verma has no need of the anti-files described by Bailey since it already includes its own delete operation. For this reason alone, using Bailey in connection with Verma does not make sense. Bailey would simply provide a redundant operation. Moreover, using Bailey in connection with Verma would require a massive amount of unnecessary changes to the Verma system. For example, Verma would have to start maintaining two separate copies of directories for the sole purpose of the delete operation, while still maintaining its current system for all other operations. This does not make sense. Logically, each reference by its nature teaches away from the other. Accordingly, a Verma-Bailey combination is non-obvious.

The Examiner also argues that Rudoff can be combined with Verma and Bailey because is would have been obvious to combine it in order to "enable usage of a working version for file modifications (i.e. working version) that is persistent as taught by Rudoff." (See Final Office Action, Pages 6-7). To incorporate Rudoff with Verma may or may not make sense. Verma is concerned with committing transactions to disk. Rudoff relates to efficient storage of disk space. For example, once Verma commits transactions to disk, it may be beneficial to efficiently store the committed data. However, this is irrelevant to Claim 1.

The Examiner also states that "[o]ne of ordinary skill in the art would be motivated to employ Bailey in order to enable a logical file structure with the capability to handle file capabilities . . . and to employ Rudoff in order to optimize storage requirements in the manipulation of working versions in the management file systems." (See Final Office Action, Pages 6-7). This may or may not make sense. Since, neither Bailey nor Rudoff teaches the aspects the Examiner cites them for the resulting combination is irrelevant to Claim 1.

For at least these reasons, the Appellant respectfully submits that the Examiner has failed to establish a prima facie case of obviousness under 35 U.S.C. 103(a), and therefore, the rejection of Claim 1 should be withdrawn.

Claim 76 is a computer-readable storage medium claim reasonably analogous to Claim 1 and is patentable for at least the same reasons discussed above.

5. Independent Claims 12, 25, 31, 87, 99, and 105

Independent claims 12 and 87 recite "recording information about one or more items in a file system to a comparison file, wherein the information recorded to the comparison file includes location information to identify where in the file system the one or more items is located" and "upon a synchronization event, comparing information in the comparison file to the working version to determine if a first working item in the working version was copied from a second working item in the working version." These limitations are similar to those recited in Claim 1, and, therefore, Claims 12 and 87 are patentable for at least the same reasons as those discussed above in connection with Claim 1.

Claims 12 and 87 further recite "**determin[ing] if a first working item in the working version was copied from a second working item in the working version.**" As noted above, none of Verma, Bailey, or Rudoff describes comparing information in the comparison file. In addition, they do not compare an item from a working version to see if it was copied from a second working item in the working version. The Examiner cites read/write operation in Verma (See Final Office Action, Page 9), but the Examiner ignores that one of the key points here is that no data has to be copied from the working version back to the original version of the file system. Put another way, the claimed invention is able to simply move information from one location on the original file system to another location without copying any data from the working version back to the original file system. This is not taught by any of the cited references.

Independent claims 25 and 99 recite "recording information about a file system to a comparison file, the information identifying at least a first item in the file system located at a first location, the first location being identifiable by a first location information" and "upon a synchronization event, comparing information in the comparison file to the working version to detect if at least one of two operations were performed on the working version, the operations including changing the first working location information for the first working item, and editing a content of the first working item." These limitations are similar to those recited in Claim 1, and, therefore, Claims 25 and 99 are patentable for at least the same reasons as those discussed above in connection with Claim 1.

Claims 25 and 99 further recite "**detecting if at least one of two operations were performed on the working version, the operations including changing the first working location information for the first working item, and editing a content of the first working item**". As noted above, none of Verma, Bailey, or Rudoff describes comparing information in the comparison file. In addition, they do not detect if at least one of two operations were performed on a working version. As a matter of fact, Bailey appears to only be able to determine if one operation has been performed (e.g., a delete). Again, the Examiner equates the read/write operation in Verma to this feature (See Final Office Action, Page 9), but the Examiner ignores that one of the key points here is that very little, if any, data has to be copied from a working version of the file system back to the original version of the file system. Put another way, the claimed invention is able to simply move information from one location on the original file system to another location without copying much, if any, data from the working version back to the original file system. This is not taught by any of the cited references.

Independent claims 31 and 105 recite "recording information about a file system to a comparison file, the information identifying at least a first item in the file system located at a first location, the first location being identified by a first location information" and "upon a synchronization event, comparing information in the comparison file to the working version to determine if a compound operation was performed on the first working item, the compound operation including at least two successive operations from a set of operations that consist of changing the first working location information for the first working item, making a first working copy from the first working item, and editing a content of the first working item." These limitations are similar to those recited in Claim 1, and, therefore, Claims 31 and 105 are patentable for at least the same reasons as those discussed above in connection with Claim 1.

Claims 31 and 105 further recite "**determining if a compound operation was performed on the first working item, the compound operation including at least two successive operations from a set of operations that consist of changing the first working location information for the first working item, making a first working copy from the first working item, and editing a content of the first working item**". As noted above, none of Verma, Bailey, or Rudoff describes comparing information in the comparison file. In addition, they do not determine if a compound operation was performed on a first working operation. The Examiner again cites read/write operation in Verma (See Final Office Action, Page 9), but the Examiner ignores does not specifically indicate which portion of Verma (or Bailey) illustrates this point. Verma, arguably, performs multiple operations on a file when it commits items to disk. However, again, Verma does not do this through the user of a comparison file or a working version of the file system. Thus, this feature is not taught by any of the cited references.

In addition, the Appellant asserts that the same motivation to combine issues exist with these independent claims as exists with Claim 1.

For at least these reasons, the Appellant respectfully submits that the Examiner has failed to establish a prima facie case of obviousness under 35 U.S.C. 103(a), and therefore, the rejection of Claims 12, 25, 31, 87, 99, and 105 should be withdrawn.

6. The Dependent Claims

Claims 2-3, 6, 9, 13-16, 21, 24, 32-33, 77-78, 81, 84, 88-90, 95, and 106-107 are dependent claims, each of which depends (directly or indirectly) on at least one of Claims 1, 12, 25, 31, 76, 87, 99, and 105. Each of these claims is therefore allowable for the reasons given above for the claim on which it depends. In addition, each of Claims -3, 6, 9, 13-16, 21, 24, 32-33, 77-78, 81, 84, 88-90, 95, and 106-107 introduces one or more additional limitations that independently render it patentable. However, due to the fundamental differences already identified, to expedite the positive resolution of this case a separate discussion of those limitations is not included at this time, although the Appellant reserves the right to further point out the differences between the cited art and the novel features recited in the dependent claims.

II. EACH OF CLAIMS 4, 5, 7, 8, 10, 11, 17-20, 22, 23, 26-30, 79, 80, 82, 83, 85, 86, 91-94, 96-98, AND 100-104 ARE PATENTABLE OVER AN ALLEGED VERMA-BAILEY-RUDOFF-WOLFF COMBINATION

Claims 4, 5, 7, 8, 10, and 11 depend from Independent Claim 1. Thus, if Claim 1 is patentable over Verma, Bailey, Rudoff, and Wolff, then it follows that its dependent claims are also patentable over those same references. Wolff does not teach or suggest "**recording information about one or more items in a file system to a comparison file**" and "**upon a synchronization event, comparing the location information for the one or more items in the comparison file to the working version to determine if any of the corresponding one or more working items has been moved to a new location in the working version**" as recited in Claim 1. Notably, the Examiner does not allege that it does. Thus, Claim 1 is patentable over an alleged Verma-Bailey-Rudoff-Wolff combination.

In addition, Claim 4 further recites "wherein generating the working version of a portion of the file system includes recording a creation time of the one or more working items in the comparison file." Wolff does describe recording a creation time (See Wolff, Col. 23, lines 20-25). However, the creation time in Wolff is not recorded with respect to one or more working items in a comparison file. Accordingly, Claim 4 is independently patentable over Verma, Bailey, Rudoff, and Wolff, individually and in any alleged combination thereof.

Claim 5 further recites "locating the one or more working items in the working version using the creation time." Wolff may describe using a creation time, however, it does not use the creation time of an item to locate one or more working items in the working version. Accordingly, Claim 5 is independently patentable over Verma, Bailey, Rudoff, and Wolff, individually and in any alleged combination thereof.

Claim 7 further recites "using the creation time of the one or more working items to determine if any of the one or more working items has been deleted from the working version and signaling the file system to delete the corresponding item in the file system." None of the references describe using the creation time of the one or more working items to determine if any of the one or more working items has been deleted." Accordingly, Claim 5 is independently patentable over Verma, Bailey, Rudoff, and Wolff, individually and in any alleged combination thereof.

Claim 8 further recites recording a modification time of the one or more working items in the comparison file." Again, Wolff may describe using a modification time, but it does not describe recording a modification time of one or more working items in a comparison file. Accordingly, Claim 8 is independently patentable over Verma, Bailey, Rudoff, and Wolff, individually and in any alleged combination thereof.

Claim 10 further recites "determining if the one or more working items were edited subsequent to making the working version using the modification time of the one or more working items." Wolff (and none of the other cited references) determines if a working item has been edited subsequent to making a working version. Accordingly, Claim 10 is independently patentable over Verma, Bailey, Rudoff, and Wolff, individually and in any alleged combination thereof.

Claim 11 further recites "determining if a subsequent modification time of the one or more working items is different than the recorded modification time for the one or more working items." Wolff (and none of the other cited references) subsequent modification time of the one or more working items is different than the recorded modification time for the one or more working items. Accordingly, Claim 11 is independently patentable over Verma, Bailey, Rudoff, and Wolff, individually and in any alleged combination thereof.

A. There is No Motivation to Combine

In each of these claims, the Examiner argues that it would have been obvious to combine Verma, Bailey, Rudoff, and Wolff. Above, the Appellant has already argued that there is no motivation to combine Verma, Bailey, and Rudoff. Moreover, there is no motivation to combine Wolff with any of the other cited references.

The Examiner argues that it would have been "obvious to one of ordinary skill in the art at the time the invention was made to modify Verma to utilize a file system management item containing a creation time parameter as taught by Wolff. One of skill in the art would be motivated to employ Wolff in order to optimize and improve distributed processing of data within a network environment." The problem with this reasoning is that the Examiner does not identify how a creation time would optimize or improve the distributed processing of data within a network environment and how that relates specifically to Verma (and the other references).

The Appellant respectfully submits that a creation time would not in fact improve Verma's processing speed. Verma executes its operations regardless of creation time. Adding a creation time would just add unnecessary overhead to the system. Similarly, Bailey does not need to keep track of creation time (and other factors), since the anti-file is sufficient to indicate when a file has been deleted. The creation time is virtually irrelevant to any other features of Bailey. Finally, Rudoff optimizes space; it moves data from one location to another. It would, ideally, maintains the original time stamp on the data when it moves it from one location to another. Otherwise, Rudoff may create inconsistencies in data that are undesirable. Accordingly, the Examiner's reasoning does not hold.

For at least this reason, Claims 4, 5, 7, 8, 10, and 11 are patentable over the alleged combination of Verma, Bailey, Rudoff, and Wolff.

B. Other dependent claims

Claims 17-20, 22, 23, 26-30, 79, 80, 82, 83, 85, 86, 91-94, 96-98, 100-104 are dependent claims, each of which depends (directly or indirectly) on Claims 12, 25, 31, 76, 87, 99, and 105. Each of these claims is therefore allowable for the reasons given above for the claim on which it depends. As with Claims 4, 5, 7, 8, 10, and 11, each of Claims 17-20, 22, 23, 26-30, 79, 80, 82, 83, 85, 86, 91-94, 96-98, 100-104 introduces one or more additional limitations that independently render it patentable.

For the reasons set forth above, Appellant respectfully submits that all pending claims are patentable over the art of record, including the art cited but not applied. Accordingly, allowance of all claims is hereby respectfully solicited.

CLAIMS APPENDIX

1. (Previously presented) A computer-implemented method for managing files, the method comprising:

recording information about one or more items in a file system to a comparison file, wherein the information recorded to the comparison file includes location information to identify where in the file system the one or more items are located;

generating a working version of a portion of the file system, the working version including at least one or more working items that correspond to the one or more items located in the file system;

persistently maintaining the working version; and

upon a synchronization event, comparing the location information for the one or more items in the comparison file to the working version to determine if any of the corresponding one or more working items has been moved to a new location in the working version.

2. (Previously presented) The computer-implemented method of claim 1, wherein if any of the corresponding working items has been moved in the working version, the method includes causing the item in the file system to move to a corresponding new location in the file system.

3. (Previously presented) The computer-implemented method of claim 1, wherein comparing the location information for the one or more items in the comparison file to the working version to determine if any of the corresponding one or more working items has been moved to a new location includes determining if any of the one or more working items has a new name.

4. (Previously presented) The computer-implemented method of claim 3, wherein generating the working version of a portion of the file system includes recording a creation time of the one or more working items in the comparison file.

5. (Previously presented) The computer-implemented method of claim 3, wherein comparing the location information for the one or more items in the comparison file to the working version to determine if any of the corresponding one or more working items has been moved to a new location includes locating the one or more working items in the working version using the creation time.

6. (Previously presented) The computer-implemented method of claim 1, further including signaling the file system to delete the item in the file system if the corresponding item has been deleted from the working version.

7. (Previously presented) The computer-implemented method of claim 4, further including using the creation time of the one or more working items to determine if

any of the one or more working items has been deleted from the working version and signaling the file system to delete the corresponding item in the file system.

8. (Previously presented) The computer-implemented method of claim 4, wherein making a working version of a portion of the file system includes recording a modification time of the one or more working items in the comparison file.

9. (Previously presented) The computer-implemented method of claim 1, further comprising determining if the one or more working items were edited subsequent to making the working version.

10. (Previously presented) The computer-implemented method of claim 8, further comprising determining if the one or more working items were edited subsequent to making the working version using the modification time of the one or more working items.

11. (Previously presented) The computer-implemented method of claim 10, further comprising determining if a subsequent modification time of the one or more working items is different than the recorded modification time for the one or more working items.

12. (Previously presented) A computer-implemented method for managing files, the method comprising:

recording information about one or more items in a file system to a comparison file, wherein the information recorded to the comparison file includes location information to identify where in the file system the one or more items is located;

making a working version of a portion of the file system, wherein the working version includes one or more working items corresponding to the one or more items in the file system; and

persistently maintaining the working version; and

upon a synchronization event, comparing information in the comparison file to the working version to determine if a first working item in the working version was copied from a second working item in the working version.

13. (Previously presented) The computer-implemented method of claim 12, further comprising causing the portion of the file system to include the first working item after determining the first working item was copied from the second working item.

14. (Previously presented) The computer-implemented method of claim 12, wherein the second working item originates from a first item in the portion of the file system.

15. (Previously presented) The computer-implemented method of claim 13, wherein the second working item is created as new after the working version is made.

16. (Previously presented) The computer-implemented method of claim 13, wherein a content of the first working item is different than a content of the second working item.

17. (Previously presented) The computer-implemented method of claim 14, wherein making a working version includes recording both a creation time and a modification time in the comparison file for the one or more working items.

18. (Previously presented) The computer-implemented method of claim 17, wherein comparing information in the comparison file to the working version to determine if a first working item in the working version was once copied from a second working item in the working version includes identifying a creation time and a modification time for the first working item.

19. (Previously presented) The computer-implemented method of claim 18, wherein comparing information in the comparison file to the working version to determine if a first working item in the working version was once copied from a second working item in the working version includes detecting that the modification time of the first working item is before the creation time of the first working item.

20. (Previously presented) The computer-implemented method of claim 19, wherein comparing information in the comparison file to the working version to determine

if a first working items in the working version was once copied from a second working item in the working version includes matching the modification time of the first working items with the modification time of the second working item.

21. (Previously presented) The computer-implemented method of claim 12, further comprising determining if the first working item was edited after being copied from the second working item.

22. (Previously presented) The computer-implemented method of claim 18, further comprising determining that the first working item is new if the creation time of the first working items is different than a creation time of all of the items identified by the comparison file, and if the modification time for the first working item is greater than or equal to creation time for the first working item.

23. (Previously presented) The computer-implemented method of claim 22, further comprising causing the portion of the file system to include the first working item after determining the first working item is new.

24. (Previously presented) The computer-implemented method of claim 12, wherein comparing information in the comparison file to the working version to determine if a first working item in the working version was once copied from a second working item in the working version includes comparing a content of the first working item to a content of the second working item.

25. (Previously presented) A computer-implemented method for managing files, the method comprising:

recording information about a file system to a comparison file, the information identifying at least a first item in the file system located at a first location, the first location being identifiable by a first location information;

making a working version of a portion of the file system, the working version including at least a first working item originating from the first item, the first working item initially being located at a first working location, the first working location corresponding to the first location information;

persistently maintaining the working version; and

upon a synchronization event, comparing information in the comparison file to the working version to detect if at least one of two operations were performed on the working version, the operations including changing the first working location information for the first working item, and editing a content of the first working item.

26. (Previously presented) The computer-implemented method of claim 25, wherein making a working version of a portion of the file system includes recording an initial modification time for the first working item in the comparison file, the initial modification time recording a last instance when the first working item was either edited or created.

27. (Previously presented) The computer-implemented method of claim 26, wherein detecting the operation of editing the content of the first working item includes subsequently determining if the initial modification time was changed.

28. (Previously presented) The computer-implemented method of claim 27, further comprising causing the first item of the file system to include the edited content of the first working item.

29. (Previously presented) The computer-implemented method of claim 25, wherein making a working version of the file system includes recording a creation time for the first working item.

30. (Previously presented) The computer-implemented method of claim 29, wherein comparing information in the comparison file to the working version to detect if at least one of two operations were performed on the working version includes locating the first working item in the working version using the creation time.

31. (Previously presented) A computer-implemented method for managing files, the method comprising:

recording information about a file system to a comparison file, the information identifying at least a first item in the file system located at a first location, the first location being identifiable by a first location information;

making a working version of a portion of the file system, the working version including at least a first working item originating from the first item, the first working item initially being located at a first working location, the first working location corresponding to the first location information;

persistently maintaining the working version; and

upon a synchronization event, comparing information in the comparison file to the working version to determine if a compound operation was performed on the first working item, the compound operation including at least two successive operations from a set of operations that consist of changing the first working location information for the first working item, making a first working copy from the first working item, and editing a content of the first working item.

32. (Previously presented) The computer-implemented method of claim 31, further comprising causing the portion of the file system to incorporate changes made by operations performed on the working version so that the first item matches the first working item.

33. (Previously presented) The computer-implemented method of claim 31, wherein when one of the at least two successive compound operations is making a first working copy from the first working item, the method further includes determining if a selected working item in the working version was once copied from the first working item.

34-75. (Canceled)

76. (Previously presented) A computer readable storage medium carrying instructions for managing files, the instructions including instructions for performing the steps of:

recording information about one or more items in a file system to a comparison file, wherein the information recorded to the comparison file includes location information to identify where in the file system the one or more items are located;

generating a working version of a portion of the file system, the working version including at least one or more working items that correspond to the one or more items located in the file system;

persistently maintaining the working version; and

upon a synchronization event, comparing the location information for the one or more items in the comparison file to the working version to determine if any of the corresponding one or more working items has been moved to a new location in the working version..

77. (Previously presented) The computer readable storage medium of claim 76, wherein if any of the corresponding working items has been moved in the working version, the method includes causing the item in the file system to move to a corresponding new location in the file system.

78. (Previously presented) The computer readable storage medium of claim 76, wherein the step of comparing the location information for the one or more items in the comparison file to the working version to determine if any of the corresponding one or more working items has been moved to a new location includes instructions for determining if any of the one or more working items has a new name.

79. (Previously presented) The computer readable storage medium of claim 78, wherein the step of generating the working version of a portion of the file system includes instructions for recording a creation time of the one or more working items in the comparison file.

80. (Previously presented) The computer readable storage medium of claim 78, wherein the step of comparing the location information for the one or more items in the comparison file to the working version to determine if any of the working items has been moved to a new location includes locating the one or more working items in the working version using the creation time.

81. (Previously presented) The computer readable storage medium of claim 76, further including instructions for signaling the file system to delete the item in the file system if the corresponding working item has been deleted from the working version.

82. (Previously presented) The computer readable storage medium of claim 79, further including instructions for using the creation time of the one or more working items to determine if any of the one or more working items has been deleted from the working version and signaling the file system to delete the corresponding first in the file system.

83. (Previously presented) The computer readable storage medium of claim 79, wherein the step of making a working version of the file system includes instructions for recording a modification time of the one or more working items in the comparison file.

84. (Previously presented) The computer readable storage medium of claim 76, further comprising instructions for determining if the one or more working items was edited subsequent to making the working version.

85. (Previously presented) The computer readable storage medium of claim 83, further comprising instructions for determining if the one or more working items was edited subsequent to making the working version using the modification time of the one or more working items.

86. (Previously presented) The computer readable storage medium of claim 85, wherein the step of using the modification time of the one or more working items

includes instructions for determining if a subsequent modification time of the one or more working items is different than the recorded modification time for the one or more working items.

87. (Previously presented) A computer readable storage medium carrying instructions for managing files on different computers, the instructions including instructions for performing the steps of:

recording information about one or more items in a file system to a comparison file, wherein the information recorded to the comparison file includes location information to identify where in the file system the one or more items is located;

making a working version of a portion of the file system, wherein the working version includes one or more working items corresponding to the one or more items in the file system; and

persistently maintaining the working version; and

upon a synchronization event, comparing information in the comparison file to the working version to determine if a first working item in the working version was copied from a first working item location in the working version.

88. (Previously presented) The computer readable storage medium of claim 87, further comprising instructions for causing the file system to copy a corresponding

first file system item to a corresponding new location in the file system after determining the first working item was copied from the first working item location in the working version.

89. (Canceled)

90. (Previously presented) The computer readable storage medium of claim 88, wherein a content of the first working item is different than a content of a second working item.

91. (Previously presented) The computer readable storage medium of claim 87 , wherein the step of making a working version includes instructions for recording both a creation time and a modification time in the comparison file for the first working item after it has been copied to a second working item location.

92. (Previously presented) The computer readable storage medium of claim 91, wherein the step of using information in the comparison file and about the working version to determine if a first working item in the working version was once copied from a first working item location to the second working item location in the working version includes instructions for identifying a creation time and a modification time for the first working item.

93. (Previously presented) The computer readable storage medium of claim 92, wherein the step of using information in the comparison file and about the working version to subsequently determine if a first working item in the working version was once copied from a first working item location to the second working item location in the working version includes instructions for detecting that the modification time of the first working item at the first working item location is before the creation time of the first working item at the second working item location.

94. (Previously presented) The computer readable storage medium of claim 93, wherein the step of using information in the comparison file and about the working version to subsequently determine if a first working item in the working version was once copied from the first working item location to the second working item location in the working version includes instructions for matching the modification time of the first working item at the first working location with the modification time of the first working item at the second working location.

95. (Previously presented) The computer readable storage medium of claim 87, further comprising instructions for performing the steps of determining if the first working item was edited after being copied.

96. (Previously presented) The computer readable storage medium of claim 92, further comprising instructions for performing the steps of determining that the

first working item is new if the creation time of the first working item is different than a creation time of all of the items identified by the comparison file, and if the modification time for the first working item at the second working item location is greater than or equal to the creation time for the first working item at the first working item location.

97. (Previously presented) The computer readable storage medium of claim 96, further comprising instructions for performing the step of causing the file system to include a corresponding item in the file system after determining the first working item is new.

98. (Previously presented) The computer readable storage medium of claim 87, wherein the step of using information in the comparison file and about the working version to subsequently determine if a first working item in the working version was once copied from a first working item location in the working version includes instructions for comparing a content of the first working item to a content of the working item at the second working item location.

99. (Previously presented) A computer readable storage medium for managing files, the computer readable medium carrying instructions including instructions for performing the steps of:

recording information about a file system to a comparison file, the information identifying at least a first item in the file system located at a first location, the first location being identified by a first location information;

making a working version of a portion of the file system, the working version including at least a first working item originating from the first item, the first working item initially being located at a first working location, the first working location corresponding to the first location information;

persistently maintaining the working version; and

upon a synchronization event, using information in the comparison file and about the working version to detect if at least one of two operations were performed on the working version, the operations including changing the first location information for the first working item, and editing a content of the first working item.

100. (Previously presented) The computer readable storage medium of claim 99, wherein the step of making a working version of a portion of the file system includes instructions for recording an initial modification time for the first working item in the comparison file, the initial modification time recording a last instance when the first working item was either edited or created.

101. (Previously presented) The computer readable storage medium of claim 100, wherein the step of comparing information in the comparison file to the working version to detect if at least one of two operations were performed on the working

version includes instructions for subsequently determining if the initial modification time was changed.

102. (Previously presented) The computer readable storage medium of claim 101, further comprising instructions for performing the step of causing the file system to include the edited content of the first working item.

103. (Previously presented) The computer readable storage medium of claim 99, wherein the step of making persistently maintaining a working version of a portion of the file system includes instructions for recording a creation time for the first working item.

104. (Previously presented) The computer readable storage medium of claim 103, wherein the step of using information in the comparison file and about the working version to detect if at least one of two operations were performed on the working version includes instructions for locating the first working item in the working version using the creation time.

105. (Previously presented) A computer readable storage medium carrying instructions for managing files, the instructions including instructions for performing the steps of:

recording information about a file system to a comparison file, the information identifying at least a first item in the file system located at a first location, the first location being identified by a first location information;

making a working version of a portion of the file system, the working version including at least a first working item originating from the first item, the first working item initially being located at a first working location, the first working location corresponding to the first location information;

persistently maintaining the working version; and

upon a synchronization event, comparing information in the comparison file to the working version to determine if a compound operation was performed on the first working item, the compound operation including at least two successive operations from a set of operations that consist of changing the first working location information for the first working item, making a first working copy from the first working item, and editing a content of the first working item.

106. (Previously presented) The computer readable storage medium of claim 105, further comprising instructions for indicating which of the operations were performed on the first working item to change the first item in the file system if the compound operation is determined to have been performed.

107. (Previously presented) The computer readable storage medium of claim 105, wherein when one of the at least two successive compound operations is making a first working copy from the first working item, the computer readable medium further includes instructions for determining if a selected working item in the working version was once copied from the first working item.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.

CONCLUSION

The rejections to Claims 1-33, 76-88, and 90-107 made under 35 U.S.C. § 103(a) lack the requisite factual and legal basis. The Appellant respectfully submits that the imposed rejections to Claims 1-33, 76-88, and 90-107 are not viable and respectfully solicit the Honorable Board to reverse each of the imposed rejections.

Respectfully submitted,
HICKMAN PALERMO TRUONG & BECKER LLP

/JosephMOlsen#58764/
Joseph M. Olsen
Reg. No. 58,764

2055 Gateway Place, Suite 550
San Jose, CA 95110-1089
(408) 414-1080
Date: August 2, 2007

Facsimile: (408) 414-1076